

SQL PREDICATE MIGRATION

This invention relates to a method of preventing unnecessary joins between tables in a database from being executed and, in particular, to a method for automatically rewriting SQL statements to achieve this.

The concept of a join between tables in a database is well known. A simple example will be described here with reference to Figure 1. This shows two tables with the names EMP and DEPT. The first table EMP lists the names of the employees of a company under the column ENAME and the number of the department for which they work under the column DEPTNO. The table DEPT has a similar column named DEPTNO in which each department number is only listed once and adjacent to this is a column entitled DNAME giving the names of the respective departments. It can be seen that a many to one relationship exists between the tables EMP and DEPT via their respective DEPTNO columns. That is to say that each value appears only once under the DEPTNO column of DEPT but can appear many times under the corresponding column of EMP. In this context, table EMP is referred to as the detail table and table DEPT is referred to as the master table.

In the table DEPT, the column DEPTNO has values that are unique in each row and this column is referred to as the primary key of the table. This primary key is typically indexed allowing fast access to each row. In table EMP, the column DEPTNO is referred to as the foreign key. This is not necessarily an indexed column although typically it is.

If it were desired to extract the names of the employees and their respective department names, then typically, a view would be defined as follows:

```
35      SELECT ENAME, DNAME FROM EMP, DEPT
      WHERE EMP.DEPTNO = DEPT.DEPTNO
```

This view may be known as EMPDEPT. Then, an SQL statement may refer to this view EMPDEPT such as:

```
SELECT ENAME, DNAME FROM EMPDEPT
```

5

This statement requires the join to be executed in order to extract the required data from both tables. However, there are instances in which it is not necessary to execute the join in order to extract the required data but nevertheless, the database still executes the join. Clearly, it is desirable to provide a method in which this unnecessary execution of joins can be prevented.

In accordance with a first aspect of the present invention, there is provided a method of rewriting a Structured Query Language (SQL) statement in order to prevent processing of a join between a master table and a detail table in a database, the join having a join condition, the method comprising the steps of:

- a. determining in the SQL statement a unique identifier to a row of the master table;
- b. equating the unique identifier to an identifier to related rows of the detail table using the join condition;
- c. producing a revised SQL statement that only refers to the detail table using the identifier to the at least one row of the detail table; and,
- d. processing the revised SQL statement.

Hence, the invention provides a method for rewriting SQL statements which refer to two tables such that the correct data can be retrieved without executing the join if that data is contained in only one of the tables.

The unique identifier to a row of the master table may be indexed. In this case, the unique identifier to a row of the master table may be a primary key.

The identifier to at least one row of the detailed table may be a foreign key.

Typically, the join condition is an equality between a column of the master table and a column of the detail table.

5 The invention will typically be provided as a computer program comprising computer program code means adapted to perform the steps of the first aspect of the invention when said program is run on a computer.

10 Further, there may be provided a computer program product comprising program code means stored on a computer readable medium for performing a method according to the first aspect of the invention when said program product is run on a computer.

15 An embodiment of the invention will now be described with reference to the accompanying drawing, Figure 1, which shows two tables in a database.

The embodiment of the invention is best described with reference to an example SQL statement, such as:

```
20      SELECT ENAME FROM EMPDEPT  
      WHERE DNAME = "R&D"
```

This refers to the view EMPDEPT defined as:

```
25      SELECT ENAME, DNAME, DEPT.DEPTNO FROM EMP, DEPT  
      WHERE EMP.DEPTNO = DEPT.DEPTNO
```

30 Since the SQL statement refers to both of the tables shown in Figure 1, the join between them, as defined in view EMPDEPT, will be executed by the database and there is an attendant cost in processing speed due to this. However, with this SQL statement it is not, in fact, necessary to execute the join if the statement is rewritten in advance. The method of the invention performs this automatically.

35 The SQL statement may be either entered manually or using a graphical user interface but, in this example, we will assume that it is being entered manually.

The first step in the method is to determine in the SQL statement a unique identifier to a row of the master table. In this instance, the master table is DEPT and table EMP is the detailed table.

5 In this case, since the SQL statement is attempting to retrieve the data and the column ENAME in table EMP only where the corresponding DNAME value equals "R&D" then the SQL statement can be rewritten to refer to the primary key of table DEPT as follows:

10

```
SELECT ENAME FROM EMPDEPT WHERE DEPTNO = 10
```

15 In fact, this change will normally speed up the processing of the SQL statement since the primary key is typically indexed thereby allowing fast access to the rows of the table via column DEPTNO.

This SQL statement can then be combined with the definition of the view EMPDEPT to produce an SQL statement as shown:

20

```
SELECT ENAME FROM EMP, DEPT
WHERE EMP.DEPTNO = DEPT.DEPTNO
AND DEPT.DEPTNO = 10
```

25 The next step in the method is to equate the unique identifier to a row of the master table to an identifier to related rows of the detail table using the join condition. In this case, the join condition is that EMP.DEPTNO = DEPT.DEPTNO and so the above SQL statement can be reduced to:

30

```
SELECT ENAME FROM EMP WHERE EMP.DEPTNO = 10
```

35 As can be seen, this new statement only refers to the detailed table, the join being eliminated and instead of filtering the results using a value from the DEPT table, the predicate has been migrated using the join condition to

the equivalent predicate related to the EMP table. As a result, the new SQL statement no longer refers to the DEPT table.

5 This new SQL statement can then be processed by the database and will be executed more quickly than the original SQL statement due to elimination of the join.

10 It is important to note that while the present invention has been described in the context of a fully functioning data processing system, those of ordinary skill in the art will appreciate that the processes of the present invention are capable of being distributed in the form of a computer readable medium of instructions and a variety of forms and that the present invention applies equally regardless of a particular type of signal bearing
15 media actually used to carry out distribution. Examples of computer readable media include recordable type media, such as floppy disks, a hard disk drive, RAM and CD-ROMs as well as transmission-type media such as digital and analogue communications links.

20